5

Boolean Algebra and Logic Gates

Introduction to Logic Gates

It has been mentioned earlier that a computer works on the theory of ON and OFF. A ON is also 1, whereas OFF is known as 0. These two states can be called hot (High) (1) and cold (Low) (0), also. The language for computer is based on these 1s and 0s, rather than on decimal symbols or alphabet.

For example, if alternate ON and OFF are recorded they can be symbolically represented as:

1 0 1 0 1 0 1 0

Such sequences ensure the beginning of a language that can be understood by both man and machine. The gate is a circuit with one or more input signals but only one output signal.

Thus, we can say that Gates are digital circuits because the input and output signals are either low or high voltages. Gates are often called logic circuits because they can be analyzed with Boolean algebra. It is very helpful to know the first 16 binary numbers for 0 and 1, they can be represented as:

Decimal Number	*Binary Number*
0	0000
1	0001
2	0010
3	0011
4	0100

5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Note that 0000 (0) is as much a number as any other number.

All electrical components of the machine also work in binary form, i.e., they operate in one of the following two states:

Magnets	are either	magnetised	or	demagnetised
Transistors	are either	conducting	or	non-conducting
Electric pulses	are either	present	or	absent
The system	is either	on	or	off
You	are either	sleep	or	awake

The basic thing to understand here is that all the information given to the machine is in binary state. It is represented either by the presence or the absence of signals.

Yes	or	No
True	or	False
On	or	Off
1	or	0

This two state system, a binary system, is applied to the computer in the form of Computer Logic Gates. The three main logic operators are:

AND

OR

NOT

And Gate

Consider the following statement.

IF Sachin scores a century AND gets 5 wickets THEN he will get the man of the match award. In this particular case, Sachin must meet both the conditions to get the man of the match award. If any one of the above condition is not met, he will not get the award.

So this can be represented as follows:

1 for conditions being met

0 for conditions not having met

OR

Let us now look at the above in another way.

IF Sachin scores a century OR gets 5 wickets THEN he will get the man of the match award.

So in this case even if one of the two conditions are met, he will get the award.

NOT Gates

Here the things are different. It can be explained as:

IF the input is 1	THEN the output is 0
IF the input is 0	THEN the output is 1
IF the input is NO	THEN the output is YES
IF the input is YES	THEN the output is NO

In other words, it reverses the signals and, for this reason, the NOT operator is called an INVERTER.

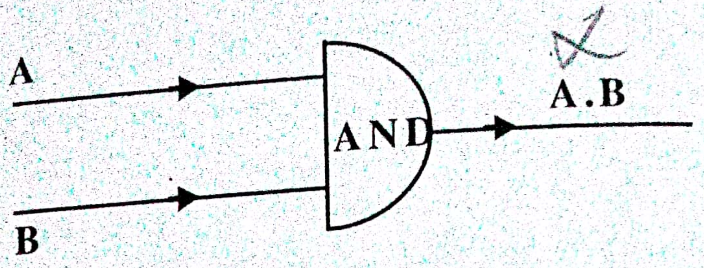
Let us see how they are represented in the Boolean Algebra.

The above mentioned AND, OR and NOT form the first category. The second category of operators are: NAND, NOR and EXOR.

AND operator

In this case two binary inputs produce an output. Both input and output are pulses. If both the inputs are 1s, the output will also be 1. It is represented by (.). Logically, it can be represented as:

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1



The following table shows all the input-output possibilities to two-input AND gates. Note that the AND gate has a high output only when both input are high. In otherwords, the AND gate is an all-or-nothing gate; a high output occurs only when all inputs are high. The AND gate is named so because of the fact that a high (1) output occurs only when A and B both input contacts are high (1).

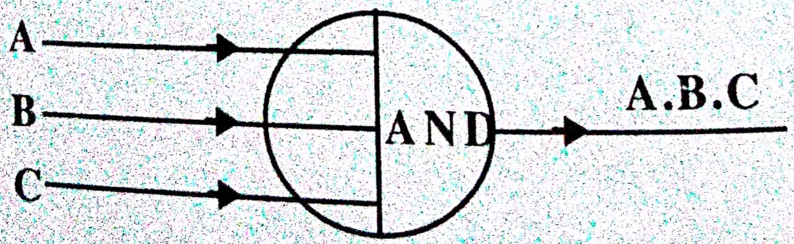
Two input AND Gate

A	B	OUT
Low (0)	Low (0)	Low (0)
Low (0)	High (1)	Low (0)
High (1)	Low (0)	Low (0)
High (1)	High (1)	High (1)

The AND gate is named so because of the fact that a high (1) output occurs only when A and B both input contacts are high (1).

More then Two Inputs

Three input AND gate can be represented as follows:



The inputs are A, B and C. When all inputs are high, all diodes are

nonconducting and the supply voltage pulls the output up to a high voltage. If all inputs are low, all diodes conduct and pull the output down to a low voltage. Even one conducting diode will pull the output down to a low voltage. The following table summarizes all input-output possibilities, in binary codes.

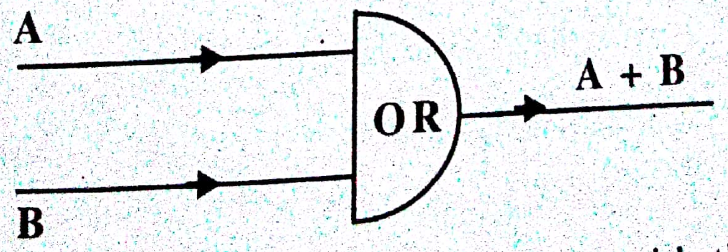
Three input AND Gate			
A	B	C	OUT
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

An AND gate can have as many inputs are needed; add one diode for each additional input. Five diodes result in a five AND gate. Seven diode result in a seven-input AND gate.

No matter, how many inputs an AND gate has, acrtion is same, i.e., all inputs must be high to get a high output.

Or operator

The OR operation produces a pulse if its inputs are pulses. It is like a parallel circuit. An expression formed with OR operator is true if either of the given proposition is true or if all the proposition are true. It is indicated by (+). Logically, it can be represented as:



A OR gate can be represented in binary form with the help of the

following table:

Two input OR Gate		
A	B	OUT
0	0	0
0	1	1
1	0	1
1	1	1

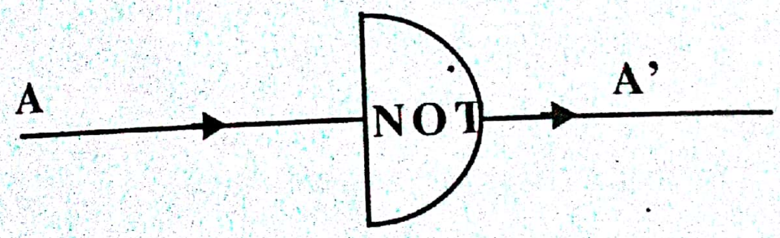
The binary 0 stands for low voltage and binary 1 for high voltage. Notice that one or more high inputs produce a high output; this is why the circuit is called an OR gate.

Inverter

NOT operator

As mentioned before, it is an inversion gate. It is unary in nature, i.e., it is performed on a single bit at a time. It changes an input to its opposite state. It is represented by as: 0's are replaced by 1's. and 1's are replaced by 0's. The NOT of A would be represented by A'. Logically, it can be represented as:

A = A'



In the tabular form NOT gates can be represented as follows:

IN	OUT
Low (0)	High (1)
High (1)	Low (0)

Or it can be represented as:

IN	OUT
Low	Low
High	High

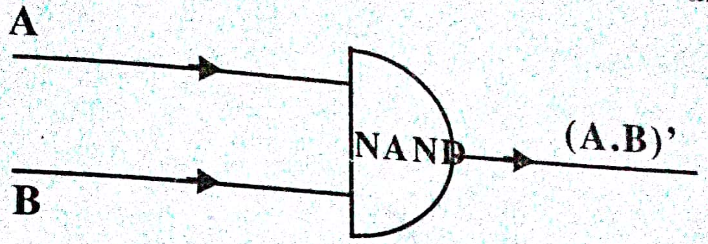
A	A'
0	1
1	0

Scanned by CamScanner

NAND operator

This is a combination of AND and NOT. An operation formed by NAND is true if either one of the two propositions is false or if both of them are false. The output is false only when both the inputs are true.

All the three gates NOT, AND and OR can be represented in terms of any NAND gates. Logically, it can be represented as:



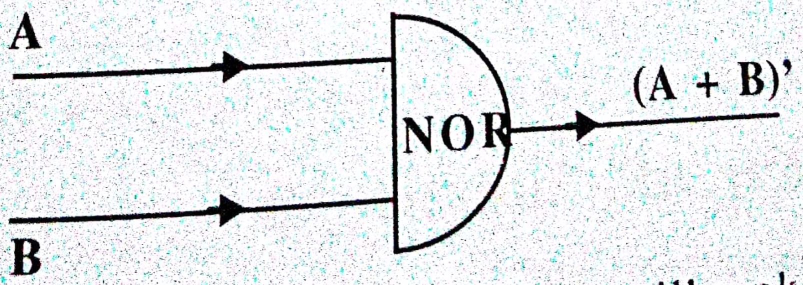
Y = (A.B)'

The following truth table representation will make it easier to understand.

Truth Table For NAND		
A	B	(A.B)'
1	0	1
0	1	1
1	1	0
0	0	1

NOR operator

It is a combination of OR followed by NOT. An expression formed by NOR is true if both the constituent propositions are false. Logically it can be represented as:



X = (A+B)'

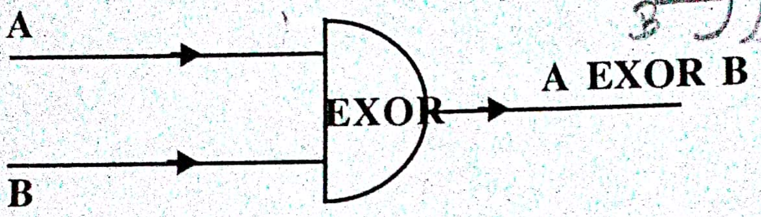
The following truth table representation will make it easier to understand.

Truth Table For NOR		
A	B	(A+B)'
1	0	0
0	1	0
1	1	0
0	0	1

EXOR operator

This is an exclusive OR operation. An expression formed with EXOR is true if either one of the two propositions is true, but it is false when both the propositions are true. It is represented as EXOR. Logically, it can be represented as:

X= A+B



The following truth table representation will make it easier to understand.

Truth Table For EXOR		
A	B	A EXOR B
1	0	1
0	1	1
1	1	0
0	0	0

Now that you have seen the various logic gates and their operations. These form the base of all the logics which are used in the making of chips, which in turn form the base of the computer chips.

Digital Integrated Circuits

A manufacturer can produce miniature circuits on the surface of a small piece of semiconductor material called a chip, by using advanced

photographic techniques. The finished network is too small, you need a microscope to see the connections. Such a circuit is called an Integrated circuit (IC) because the component (transistors, diode, resisters) are an integral part of the chip.

Large-scale integration (LSI) refers to more than 100 gates per chip. Medium-scale integration (MSI) means 12 to 100 gates per chip and small-scale integration (SSI) refers the ICs with fewer than 12 on the same chip.

The two basic techniques for:manufacturing ICS are bipolar and metal-oxide semiconductor (MOS). The first fabricates bipolar transistors on chip; the second, MOS field-effect transistors (MOSFETs). Bipolar technology is preferred for SSI and MSI because it is faster. MOS technology dominates the LSI field because more MOSFETs can be packed into the same chip area.

A digital family is a group of compatible devices with the same logic levels and supply voltages. The word compatible means that you can connect the output of one device to the input of another. With the devices of a digital family, you can create a wide variety of logic circuits.

These families are in the bipolar category:

DTL	Diode-transistor logic
TTL	Transistor logic
ECL	Emitter - coupled logic

DTL uses diodes and transistors; this design was very popular but now it is obsolete. TTL was introduced in 1964 by Texas instruments. TTL is a widely used family of digital devices. TTL is fast. inexperience, and easy to use.

TTL uses transistors almost exclusively. It has become the most popular family of SSI and MSI chips. ECL is the fastest logic family and is used in high-speed applications. These families are in the MOS category:

PMOS	p-channel MOSFETS
NMOS	n-channel MOSFETS

CMOs complementary MOSFETS

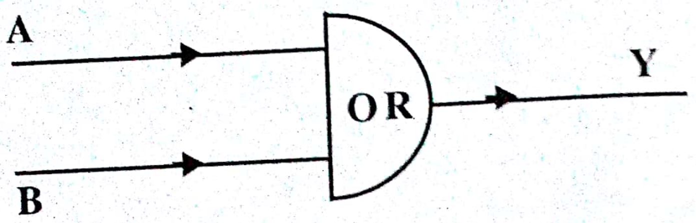
PMOS is the oldest and slowest type. It is now becoming obsolete. NMOS dominates the LSI-field in its use for microprocessors and memories. CMOS is a push-pull management of n-channel and p-channel MOSFETS.

CMOS is used in digital wristwatches, pocket calculator, etc. In other word, CMOS is used where low power consumption is needed.

De Morgan's First Theorem

The following figure shows a two-input NOR gate and the Boolean equation for NOR gate is:

$$Y = \overline{A + B}$$



The truth table for two-input NOR gate is shown below.

Two-input NOR gate.		
A	B	$\overline{A + B}$
0	0	1
1	0	0
0	1	0
1	1	0

The truth table for bubbled AND gate is shown in the following table.

Two-input bubbled AND gate		
A	B	$\overline{A}\,\overline{B}$
0	0	1
0	1	0
1	0	0
1	1	0

The first equation described a NOR gate and the second equation described a bubbled AND gate. Since the outputs are equal for the same inputs, we can equate them

$$\overline{A + B} = \overline{A}\ \overline{B}$$

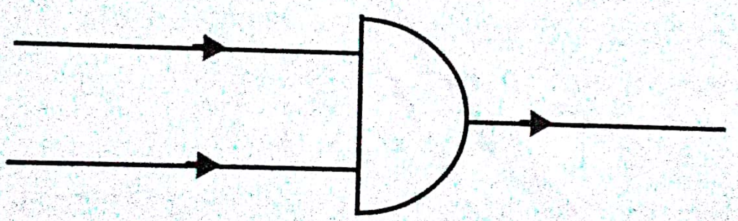
This identity is known as De Morgan's first theorem. It says the complement of a sum equals the product of the complements.

De Morgan's Second Theorem

The complement of a logical product equals the logical sum of the complements. In terms of circuits, a NAND gate is equivalent to a bubbled Or gate.

The following figure shows the two-input NAND gate and the Boolean equation for NAND gate is

$$Y = \overline{A\,B}$$

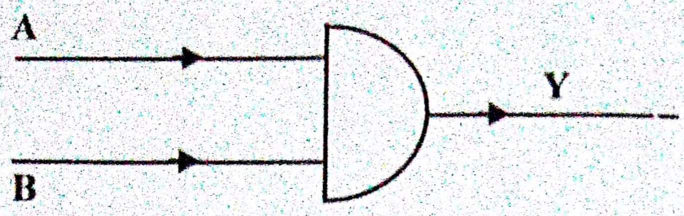


The truth table for two-input NAND gate is described below.

Two input NAND gate		
A	**B**	**A B**
0	0	1
0	1	1
1	0	1
1	1	0

The following figure shows the OR gate with inverted inputs (bubbled OR gate) and the Boolean equation for this is:

$$Y = \overline{A} + \overline{B}$$



The truth table for bubbled OR gate is described below:

Two-input Bubbled OR gate		
A	B	$\overline{A} + \overline{B}$
0	0	1
0	1	1
1	0	1
1	1	0

The first equation describes a NAND gate and the second equation describes a bubbled OR gate. Since the outputs are equals for the same inputs, we equate them

$$\overline{AB} = \overline{A} + \overline{B}$$

This is known as De Morgan's second theorem. It says the complement of a logical product equals the logical sum of the complements. A NAND gate is equivalent to a bubbled Or gate. So you can interchange the circuit very often to reduce complicated circuits to simpler forms.

Karnaugh Maps

A Karnaugh map is a graphical display of the fundamental products in a truth table. Many engineers don't simplify equation with Boolean algebra, they use a method based on Karnaugh maps. First we will discuss how to construct a Karnaugh map for a given truth table.

Two Variable Map

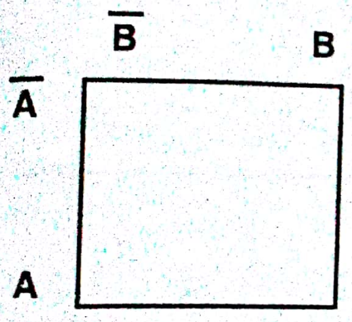
Suppose you have a truth table, involving two variables as shown in the following table.

A	B	Y
0	0	0

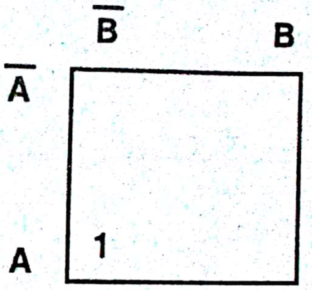
0	1	0
1	0	1
1	1	1

As there are only two variable hence one variable represents the rows and second variable represents the columns. Note the order of the variables and their complements; the vertical column has A followed by A and the horizontal row has B followed by B as shown in the following figure (a).

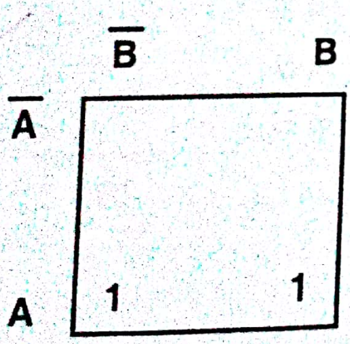
Find the output in the truth table. The first high (1) output to appear in for the input of A = 1 and B = 0. The fundamental product for this is AB. Now, enter 1 in the Karnaugh map is shown in the following figure (b).



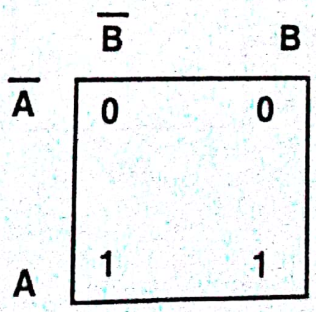
(a)



(b)



(c)



(d)

Similarly, find the other output 1 in the truth table. The next 1 output to appear is for the input A = 1 and B = 1. The fundamental product for this is AB. Now place the 1 at their corresponding place as shown in above figure (c). After placing the all 1s on the karnaugh map, enter all 0s in the renaming spaces as shown in the above figure (d).

Three Variable Map

Suppose you have a truth table involving three variables as shown in the following table. It is especially important to notice the order of the variables and their complements. The vertical column is labeled A B, AB, AB and AB. This order is not a binary progression; instead it follows the order of 00, 01, 11 and 10.

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1 --> A B C
0	1	1	1 --> A B C
1	0	0	0
1	0	1	0
1	1	0	0

	$\overline{C}$	C
$\overline{A}\,\overline{B}$		
$\overline{A}B$		
AB		
$A\overline{B}$		

(a)

	$\overline{C}$	C
$\overline{A}\,\overline{B}$		
$\overline{A}B$	1	1
AB		1
$A\overline{B}$		

(b)

	$\overline{C}$	C
$\overline{A}\,\overline{B}$	0	0
$\overline{A}B$	1	1
AB	0	1
$A\overline{B}$	0	0

(c)

Now, look the output 1s in the above table. The fundamental products of the variable are ABC, ABC, and ABC. Place these 1s in Karnaugh map are shown in the figure (b). The final step is to enter 0s in the remaining space figure (c). This way we get the fundamental products needed for the sum-of-products circuit.

Four Variable Map

Many MSI circuits process binary words of 4 bits called nibbles. For this reason, logic circuits are often designed to handle four variables. Suppose you have the following truth table.

A	B	C	S	Y
0	0	0	0	0
0	0	0	1	1 --> A B C D
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1 --> A B C D
0	1	1	1	1 --> A B C D
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1 --> A B C D
1	1	1	1	0

To obtain the Karnaugh map of four variables, the variables are divided into two groups each containing two variables. The two variables are drawn as rows while other two variables represents the column of the graph. There is no restriction to choose the group for rows and columns but the concatenation of the variables to represents the fundamental

product of variable must be in the same order. The vertical column is labeled A B, AB, A B and AB. The horizontal row is labeled C D, CD, C D and CD.

Now, look the output 1s in above table. The fundamental products of variable are A B C D, A B C D, A B C D and A B C D. Place these 1s in Karnaugh map as shown in the following figure (b). The final step is to enter 0s in the remaining space figure (c). This represents the four variable Karnaugh map.

	$\overline{C}\,\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\,\overline{B}$				
$\overline{A}B$				
AB				
$A\overline{B}$				

(a)

	$\overline{C}\,\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\,\overline{B}$		1		
$\overline{A}B$			1	1
AB				1
$A\overline{B}$				

(b)

	$\overline{C}\,\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\,\overline{B}$	0	1	0	1
$\overline{A}B$	0	0	1	1
AB	0	0	0	1
$A\overline{B}$	0	0	0	0

(c)

Let's Revise

1. A computer wo OFF is known
2. In AND opera
3. The OR opera
4. NOT operato
5. NAND opera
6. NOR operato
7. EXOR oper
8. The two ba oxide semi
9. TTL uses
10. CMOS is a
11. A Karnaug truth table
12. De Morga product o
13. De Morga equals th

Answer t

1. What a
2. What i
3. Define
4. Descr
5. Defin
6. Defin
7. Defi
8. Defi
9. Defi

ra and Logic Gates
ertical column is
is labeled C D,

mental products
C D. Place these
e (b). The final
This represents

Let's Revise

1. A computer works on the theory of ON and OFF. A ON is also 1, whereas OFF is known as 0
2. In AND operator two binary inputs produce an output.
3. The OR operation produces a pulse if its inputs are pulses.
4. NOT operator is an inversion gate.
5. NAND operator is a combination of AND and NOT.
6. NOR operator is a combination of OR followed by NOT.
7. EXOR operator is an exclusive OR operation.
8. The two basic techniques for:manufacturing ICS are bipolar and metal-oxide semiconductor (MOS).
9. TTL uses transistors almost exclusively.
10. CMOS is a push-pull management of n-channel and p-channel MOSFETS.
11. A Karnaugh map is a graphical display of the fundamental products in a truth table.
12. De Morgan's first theorem says that the complement of a sum equals the product of the complements.
13. De Morgan's second theorem says that the complement of a logical product equals the logical sum of the complements.

Answer the Following Questions

1. What are logic gates?
2. What is an And Gate?
3. Define OR and NOR Gates.
4. Describe NOR operation.
5. Define AND gate for more than two inputs
6. Define OR operator.
7. Define NOT operator.
8. Define the truth table of NOT operator
9. Define EXOR operator